In the Claims:

Please amend claims 1, 6-8, 10, 19-20, 22, 27-28, 30, 32, 34-35, 37-39, 41, 43 and 45-53, as indicated below.

1. (Currently amended) A computer-implemented method for the secure exchange of objects in a distributed computing environment, comprising:

a user accessing a client device;

generating a computer programming language object from a data ~~representation~~ markup language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client device;

the client device receiving a message in the data ~~representation~~ markup language from a service device in the distributed computing environment prior to said generating a computer programming language object, wherein the message includes the data ~~representation~~ markup language representation of the object;

the user terminating said accessing the client device; and

deleting the computer programming language object in response to said terminating access, wherein the deleted object is not accessible for use by subsequent users of the client device.

2. (Canceled)

3. (Original) The method as recited in claim 1, wherein said accessing a client device comprises the user coupling an identification device to the client device, wherein

the identification device provides identification information of the user to the client device, and wherein said terminating said accessing comprises decoupling the identification device from the client device.

4. (Original) The method as recited in claim 3, wherein the identification device is a smart card.

5. (Original) The method as recited in claim 1, wherein said accessing a client device comprises the user logging on to the client device by providing user identification to the client device, and wherein said terminating said accessing comprises the user logging off the client device.

6. (Currently amended) The method as recited in claim 1, wherein said generating a computer programming language object from a data ~~representation~~ markup language representation of the object is performed by a virtual machine executing within the client device.

7. (Currently amended) The method as recited in claim 1, further comprising:

generating a plurality of computer programming language objects from data ~~representation~~ markup language representations of the objects; and

deleting the plurality of computer programming language objects in response to said terminating access.

8. (Currently amended) The method as recited in claim 1, wherein said data ~~representation~~ markup language is eXtensible Markup Language (XML).

9. (Original) The method as recited in claim 1, wherein said computer programming language is the Java programming language.

10. (Currently amended) A computer-implemented method for the secure exchange of objects in a distributed computing environment, comprising:

a user accessing a client device;

the client device receiving a message in a data ~~representation~~ markup language from a service device in the distributed computing environment, wherein the message includes a data ~~representation~~ markup language representation of a computer programming language object;

determining if the user has access rights to the computer programming language object;

if said determining determines the user has access rights to the computer programming language object, generating the object from the data ~~representation~~ markup language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client device; and

if said determining determines the user does not have access rights to the computer programming language object, not generating the object.

11. (Original) The method as recited in claim 10, wherein the message further includes access information for the computer programming language object, wherein said determining if the user has access rights to the computer programming language object uses the access information.

12. (Original) The method as recited in claim 10, further comprising:

the user terminating said accessing the client device; and deleting the computer programming language object in response to said terminating access, wherein the deleted object is not accessible for use by subsequent users of the client device.

13. (Original) The method as recited in claim 12, wherein said accessing a client device comprises the user coupling an identification device to the client device, wherein the identification device provides identification information of the user to the client device, and wherein said terminating said accessing comprises decoupling the identification device from the client device.

14. (Original) The method as recited in claim 13, wherein the identification device is a smart card.

15. (Original) The method as recited in claim 12, wherein said accessing a client device comprises the user logging on to the client device by providing user identification to the client device, and wherein said terminating said accessing comprises the user logging off the client device.

16. (Original) The method as recited in claim 10, further comprising:

the user terminating said accessing the client device; and

storing the computer programming language object in response to said terminating access.

17. (Original) The method as recited in claim 16, further comprising:

the user accessing the client device subsequent to said storing the object; and

accessing the stored object during said accessing the client device.

18. (Original) The method as recited in claim 17, further comprising storing access rights information of the user with the object, wherein said accessing the stored object comprises verifying the access rights of the user with the stored access rights information.

19. (Currently amended) The method as recited in claim 10, wherein said generating a computer programming language object from a data ~~representation~~ markup language representation of the object is performed by a virtual machine executing within the client device.

20. (Currently amended) The method as recited in claim 10, wherein said data ~~representation~~ markup language is eXtensible Markup Language (XML).

21. (Original) The method as recited in claim 10, wherein said computer programming language is the Java programming language.

22. (Currently amended) A device, comprising ~~configured to~~:

a processor; and

a memory coupled to the processor, wherein the memory stores program instructions executable by the processor to:

accept user input to initiate user access of the device;

generate a computer programming language object from a data ~~representation~~ markup language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the device;

receive a message in the data ~~representation~~ markup language from a source prior to said generating a computer programming language object, wherein the message includes the data ~~representation~~ markup language representation of the object;

terminate said user access; and

delete the computer programming language object in response to said terminating access;

wherein the deleted object is not accessible for use by subsequent users of the device.

23. (Canceled)

24. (Original) The device as recited in claim 22, wherein the device is further configured to:

couple to an identification device, wherein the identification device is configured to provide identification information of the user to the device during said accepting user input; and

decouple from the identification device subsequent to said generating, wherein the device performs said terminating said user access in response to said decoupling.

25. (Original) The device as recited in claim 24, wherein the identification device is a smart card.

26. (Original) The device as recited in claim 22, wherein the device is further configured to accept user input to initiate said terminating said user access.

27. (Currently amended) The device as recited in claim 22, wherein the device is further configured to:

generate a plurality of computer programming language objects from data ~~representation~~ markup language representations of the objects; and

delete the plurality of computer programming language objects in response to said terminating access.

28. (Currently amended) The device as recited in claim 22, comprising:

~~a processor;~~

~~a memory;~~

a virtual machine executed by said processor from said memory, wherein said generating is performed by the virtual machine.

29. (Original) The device as recited in claim 28,

wherein said accepting, said terminating, and said deleting are performed by the virtual machine, wherein the object is stored in the memory subsequent to said generating, and wherein, in said deleting, the object is deleted from the memory.

30. (Currently amended) The device as recited in claim 22, wherein said data ~~representation~~ markup language is eXtensible Markup Language (XML).

31. (Original) The device as recited in claim 22, wherein said computer programming language is the Java programming language.

32. (Currently amended) A distributed computing system, comprising:

a client hardware device; and

a service hardware device;

wherein the client hardware device is configured to:

accept user input to initiate user access of the device;

receive a message in a data ~~representation~~ markup language from the service hardware device, wherein the message includes a data ~~representation~~ markup language representation of a computer programming language object;

determine if the user has access rights to the computer programming language object;

if said determining determines the user has access rights to the computer programming language object, generate the object from the data ~~representation~~ markup language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client hardware device; and

if said determining determines the user does not have access rights to the computer programming language object, not generate the object.

33. (Original) The system as recited in claim 32, wherein the message further includes access information for the computer programming language object, wherein said determining if the user has access rights to the computer programming language object uses the access information.

34. (Currently amended) The system as recited in claim 32, wherein the client hardware device is further configured to:

accept user input to terminate said access of the client hardware device; and

delete the computer programming language object in response to said terminating said access, wherein the deleted object is not accessible for use by subsequent users of the client hardware device.

35. (Currently amended) The system as recited in claim 34, wherein the client hardware device is further configured to:

couple to an identification device, wherein the identification device is configured to provide identification information of the user to the client hardware device during said accepting user input; and

decouple from the identification device subsequent to said generating, wherein the device performs said terminating said user access in response to said decoupling.

36. (Original) The system as recited in claim 35, wherein the identification device is a smart card.

37. (Currently amended) The system as recited in claim 32,

wherein the client hardware device comprises:

a memory; and

wherein the client hardware device is further configured to:

accept user input to terminate said access of the client hardware device; and

store the computer programming language object to the memory in response to said terminating access.

38. (Currently amended) The system as recited in claim 37, wherein the client hardware device is further configured to:

accept user input to initiate a second user access of the client hardware device subsequent to said storing the object; and

provide access to the stored object during said second user access of the client hardware device.

39. (Currently amended) The system as recited in claim 38, wherein the client hardware device is further configured to:

store access rights information of the user with the object in the memory; and

verify the access rights of the user with the stored access rights information prior to said providing access to the stored object.

40. (Previously presented) The device as recited in claim 32, comprising:

a processor;

a memory;

a virtual machine executed by said processor from said memory, wherein said generating is performed by the virtual machine, and wherein the object is stored in the memory subsequent to said generating.

41. (Currently amended) The system as recited in claim 32, wherein said data ~~representation~~ markup language is eXtensible Markup Language (XML).

42. (Original) The system as recited in claim 32, wherein said computer programming language is the Java programming language.

43. (Currently amended) A ~~tangible,~~ computer accessible storage medium ~~comprising~~ storing program instructions, wherein the program instructions are computer-executable to implement:

a user accessing a client device;

generating a computer programming language object from a data ~~representation~~ markup language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client device;

receiving a message in the data ~~representation~~ markup language from a service device in the distributed computing environment prior to said generating a computer programming language object, wherein the message includes the data ~~representation~~ markup language representation of the object;

the user terminating said accessing the client device; and deleting the computer programming language object in response to said terminating access, wherein the deleted object is not accessible for use by subsequent users of the client device.

44. (Canceled)

45. (Currently amended) The ~~tangible,~~ computer accessible storage medium as recited in claim 43, wherein the program instructions are further computer-executable to implement:

generating a plurality of computer programming language objects from data ~~representation~~ markup language representations of the objects; and

deleting the plurality of computer programming language objects in response to said terminating access.

46. (Currently amended) The ~~tangible,~~ computer accessible storage medium as recited in claim 43, wherein said data ~~representation~~ markup language is eXtensible Markup Language (XML), and wherein said computer programming language is the Java programming language.

47. (Currently amended) A ~~tangible,~~ computer accessible storage medium ~~comprising~~ storing program instructions, wherein the program instructions are computer-executable to implement:

a user accessing a client device;

the client device receiving a message in a data ~~representation~~ markup language from a service device in the distributed computing environment, wherein the message includes a data ~~representation~~ markup language representation of a computer programming language object;

determining if the user has access rights to the computer programming language object;

if said determining determines the user has access rights to the computer programming language object, generating the object from the data ~~representation~~ markup language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client device; and

if said determining determines the user does not have access rights to the computer programming language object, not generating the object.

48. (Currently amended) The ~~tangible,~~ computer accessible storage medium as recited in claim 47, wherein the message further includes access information for the computer programming language object, wherein said determining if the user has access rights to the computer programming language object uses the access information.

49. (Currently amended) The ~~tangible,~~ computer accessible storage medium as recited in claim 47, wherein the program instructions are further computer-executable to implement:

the user terminating said accessing the client device; and

deleting the computer programming language object in response to said terminating access, wherein the deleted object is not accessible for use by subsequent users of the client device.

50. (Currently amended) The ~~tangible,~~ computer accessible storage medium as recited in claim 49, wherein said accessing a client device comprises the user coupling an

identification device to the client device, wherein the identification device provides identification information of the user to the client device, and wherein said terminating said accessing comprises decoupling the identification device from the client device.

51. (Currently amended) The ~~tangible,~~ computer accessible storage medium as recited in claim 47, wherein the program instructions are further computer-executable to implement:

terminating said accessing the client device; and

storing the computer programming language object in response to said terminating access; and

storing access rights information of the user with the object.

52. (Currently amended) The ~~tangible,~~ computer accessible storage medium as recited in claim 51, wherein the program instructions are further computer-executable to implement:

the user accessing the client device subsequent to said storing the object;

verifying the access rights of the user with the stored access rights information; and

if said verifying verifies the access rights of the user to the object, allowing access to the stored object during said accessing the client device.

53. (Currently amended) The ~~tangible,~~ computer accessible storage medium as recited in claim 47, wherein said data ~~representation~~ markup language is eXtensible Markup Language (XML), and wherein said computer programming language is the Java programming language.